CLAIMS

What is claimed is:

1. A method of generating a fast-path response to a packet received onto a network interface device, the packet being received over a TCP/IP network connection, the TCP/IP network connection being identified at least in part by a TCP source port, a TCP destination port, an IP source address, and an IP destination address, the method comprising:

on the network interface device examining the packet and determining from the packet the TCP source port, the TCP destination port, the IP source address, and the IP destination address;

accessing a template header stored on the network interface device, the template header having TCP fields and IP fields;

employing a finite state machine that implements both TCP protocol processing and IP protocol processing to fill in the TCP fields and IP fields of the template header, wherein the finite state machine does not realize a TCP protocol processing layer and a discrete IP protocol processing layer where the TCP and IP layers are executed one after another in sequence, but rather the finite state machine covers both TCP and IP protocol processing; and

transmitting the fast-path response from the network interface device, the filled in template header forming at least a part of the fast-path response.

2. The method of Claim 1, wherein the network interface device comprises a processor, the finite state machine being a software state machine executing on the processor.

3. The method of Claim 1, wherein one of the TCP fields is a TCP header checksum, and wherein one of the IP fields is an IP header checksum, wherein the finite state machine updates both the TCP header checksum and the IP header checksum.

4. The method of Claim 1, wherein the network interface device is coupled to a host, the host executing a protocol stack, the protocol stack comprising a TCP protocol processing layer and an IP protocol processing layer, the method further comprising:

receiving a slow-path packet from the host, the slow-path packet having been generated by the protocol stack of the host, the slow-path packet including a transport protocol header that is not a TCP header; and

slow-path transmitting the slow-path packet from the network interface device.

5. The method of Claim 1, wherein the network interface device is coupled to a host, the host executing a protocol stack, the protocol stack comprising a TCP protocol processing layer and an IP protocol processing layer, the method further comprising:

receiving a slow-path packet from the host, the slow-path packet having been generated by the protocol stack of the host, the slow-path packet including a network protocol header that is not an IP header; and

slow-path transmitting the slow-path packet from the network interface device.

6. The method of Claim 1, wherein the network interface device comprises a processor and a transmit device, the finite state machine being a software state machine that executes on the processor, wherein the processor forms the fast-path response and causes the response to be stored in a memory, and wherein the processor causes a buffer descriptor that points to the fast-path response to be placed onto a transmit queue, and wherein the transmit device retrieves the buffer descriptor from the transmit queue and causes the fast-path response to be transmitted from the network interface device.

7. The method of Claim 6, wherein the transmit device comprises a sequencer.

8. The method of Claim 6, wherein the network interface device employs a first transmit queue and a second transmit queue, the first transmit queue being of a higher transmission priority than the second transmit queue.

9. The method of Claim 1, wherein a transmit device on the network interface device causes the fast-path response to be transmitted from the network interface device, the transmit device retrieving buffer descriptors from a first transmit queue and from a second transmit queue, the first transmit queue having a higher transmission priority than the second transmit queue.

10. The method of Claim 9, wherein the transmit device is a sequencer.

11. The method of Claim 1, wherein the finite state machine fills in all the TCP fields and all the IP fields in the template header.

12. A method of generating a fast-path response to a packet received onto a network interface device, the packet being received over a TCP/IP network connection, the TCP/IP network connection being identified at least in part by a TCP source port, a TCP destination port, an IP source address, and an IP destination address, the method comprising:

on the network interface device examining the packet and determining from the packet the TCP source port, the TCP destination port, the IP source address, and the IP destination address;

accessing a template header stored on the network interface device, the template header having TCP fields and IP fields;

filling in the TCP fields and IP fields of the template header using the determined TCP source port, TCP destination port, IP source address, and IP destination address, the TCP fields and IP fields being filled in without passing the template header down from any TCP protocol processing stack layer to any IP protocol processing stack layer; and

transmitting the fast-path response from the network interface device, the filled in template header forming at least a part of the fast-path response.

13. A method for generating a third TCP packet from a network interface device, the network interface device being coupled to a network and being coupled to a host, the host executing a network protocol stack comprising a plurality of network processing protocol layers, the network interface device comprising a processor, the method comprising:

receiving a first TCP packet onto the network interface device from the network, the first TCP packet being received via a first network connection;

slow-path processing the first TCP packet such that the stack performs substantial TCP layer processing on the first TCP packet;

receiving a second TCP packet onto the network interface device from the network, the second TCP packet being received via a second network connection, the second network connection being a TCP/IP connection; and fast-path processing the second TCP packet such that the stack of the host performs substantially no TCP layer processing on the second TCP packet, wherein in response to said receiving of the second TCP packet the network interface device generates the third TCP packet, the generating of the third TCP packet comprising:

accessing a template header stored on the network interface device, the template header having TCP fields and IP fields; and

using the processor to fill in the TCP fields and the IP fields in the template header to form the third TCP packet without passing the third TCP packet or the template header down to any lower protocol processing layer in any protocol stack.

14. The method of Claim 13, wherein the processor does not execute a network protocol stack comprising a TCP protocol processing layer and an IP protocol processing layer, but rather the processor executes software that implements a finite state machine, the finite state machine performing both TCP and IP protocol processing tasks.

15. The method of Claim 14, wherein the network interface device further comprises a transmit device, the fast-path processing further comprising:

placing a pointer to the third TCP packet into a first transmit queue; and

using the transmit device of the network interface device to retrieve the pointer from the first transmit queue, and to cause the third TCP packet to be output from the network interface device and onto the network.

16. The method of Claim 15, wherein the transmit device also retrieves pointers from a second transmit queue, one of the first transmit queue and the second transmit queue having a higher transmission priority than the other.

17. The method of Claim 16, wherein the transmit device is a sequencer.